Nintendo GameCube™ Controller Guidelines

Version 1.4

Table of Contents

DOL-06-0027-001-A2
Released: 11/1/01

Revision History

Version	Revision Date	Description of Revisions
1.4	10/19/01	- Added paragraph 1.6 “Function Return Values”
1.3	8/3/01	- Revised references in paragraph 1.1 - Changed usable range to unusable area in paragraph 1.3.4.1 and 1.3.4.2 - Revised paragraphs 1.5.1 and 1.5.2 - Revised Chapter 2 (WAVEBIRD)
1.0	7/13/01	- Released by NOA

1. The Controller

This document provides explanations and cautions regarding the Nintendo GameCube™ Controller (Controller) and the Nintendo GameCube™ Wireless Controller (WAVEBIRD).

1.1 Hot insert/removal

Hot insert/removal refers to the act of removing and inserting the Controller or the Nintendo GameCube™ Wireless Receiver (Receiver) plug while the power is turned ON to the Nintendo GameCube™. The application must support hot insert/removal.

By support for hot insert/removal, what is meant is that the application should always check each Nintendo GameCube™ Controller socket to see whether a Controller is inserted. When a Controller is inserted into a socket that previously was empty, the application should enable normal operation of the Controller that is using that socket.

The way to support hot insert/removal is to call the PADReset function for any Controller socket for which the PADRead function returns the error value PAD_ERR_NO_CONTROLLER. The application should perform this support for hot insert/removal in every frame.

There are three reasons why it is important to support hot insert/removal:

- Because a Controller might be removed and inserted while the application is running.
- To accommodate the Nintendo GameCube™ Controller's recalibration (refer to 1.2.1).
- To accommodate the switching of the channel used by the WAVEBIRD on hot insert/removal (refer to 2.1.2).

A player may intentionally remove or insert a plug while the application is running, but that is not the only time when this can happen. It can also occur when the Controller is pulled in a way that places a load on the plug, and when there is a bad connection between the plug and the socket.

1.2 Resetting the origin

The Controller and WAVEBIRD both have Control Stick, C Stick, R Button and L Button. In each case, the amount of operation is determined by the difference between the analog value and the origin value. The origin values are obtained automatically by the hardware, but it is also possible to reset the origin values from the application. These operations are explained separately below.

1.2.1 Hardware reset of origin

The resetting of numerical values, etc. that accompany these actions are performed automatically by the library, so the application (the programmer) does not need to be aware of them.

However, the timing at which the origin is reset by the hardware differs for the Controller and the WAVEBIRD.

Nintendo GameCube™ Controller

- When power is applied to Controller
  The Controller's origin is reset when power is applied to the Nintendo GameCube™ if the Controller is already plugged in, or when the Controller is plugged in to a socket if the Nintendo GameCube™ is already turned on.
- Origin Reset command
  The Controller's origin also can be reset by pressing the X Button, Y Button and START/PAUSE Button simultaneously, and holding them down for 3 seconds.

  <CAUTION>
  The developer must design the game program in such a way that the X Button, Y Button, and START/PAUSE Button will not be pressed simultaneously as a part of normal game play.

DOL-06-0027-001-A2
Released: 11/1/01

WAVEBIRD

- When power is applied to the WAVEBIRD
  The origin is set when power is turned off and back on. The new setting is not reflected in the Nintendo GameCube™ until after the receiver has received a signal from the WAVEBIRD.

1.2.2 Resetting origin from application

The Controller's origin value can also be reset from the application by using the PADRecalibrate function. This function can reset the origin of the Controller but it cannot reset the origin of the WAVEBIRD.

When the Nintendo GameCube™ RESET Button is pressed, it is necessary to reset the origin of the Nintendo GameCube™ Controller. Reset the origin from the application, regardless of whether the controller being used is a Nintendo GameCube™ Controller or a WAVEBIRD.

1.3 Control Stick, C Stick, R Button and L Button

1.3.1 Input value range

Values from +127 to -127 are accepted as input from each stick. Values from 0 to 255 are accepted as input from the R and L Buttons. If the origin is skewed, values inside this range may be input. Be sure that normal operation occurs for any value that is input inside this range.

1.3.2 Setting lower limit values

The Control Stick, C Stick, R Button and L Button do not always return to their exact origins when the player releases them. The skewing can differ from Controller to Controller, and even for the same Controller each time it is used. To prevent this skewing form being recognized as input, you should implement measures such as setting a lower limit value for play.

There will be times when the user intends to tilt the stick exactly to the side but actually tilts it a bit up or down, and times when the user intends to tilt the stick exactly up or exactly down but actually tilts it a bit to the left or right. To prevent this type of unintentional skew in control, you should implement measures such as setting a lower limit value to the perpendicular component of the direction of control when the sticks are operated straight to the sides or straight up or down.

1.3.3 Setting upper limit values

The resolution of these sticks and buttons will vary between Controllers. Moreover, the upper limit value that can be input for any given Controller will decline with age and other factors. For these reasons, the upper limit value that can be input will differ from Controller to Controller. Please do not make use of values that cannot be input.

The shape of the frame of the Controller puts restrictions on the various input values. Make sure that operations will not be effected by a change in the shape of the frame of the Controller, that enables a wider range of input and changes the maximum value.

1.3.4 Usable input value ranges and adjustments

Figure 1 shows the recommended ranges of useable values for input from the Controller. Do not use values in your applications that are outside of the range defined by the twelve-sided polygon shown in the figure.

1.3.4.1 Control Stick and C Stick

For the Control Stick and C Stick, set the usable range as shown in Figure 1 and clamp the unusable area (because of mechanical play (unshaded blocks within the polygons of Figure 1)) and all values that fall outside of the range defined by the twelve-sided polygons.

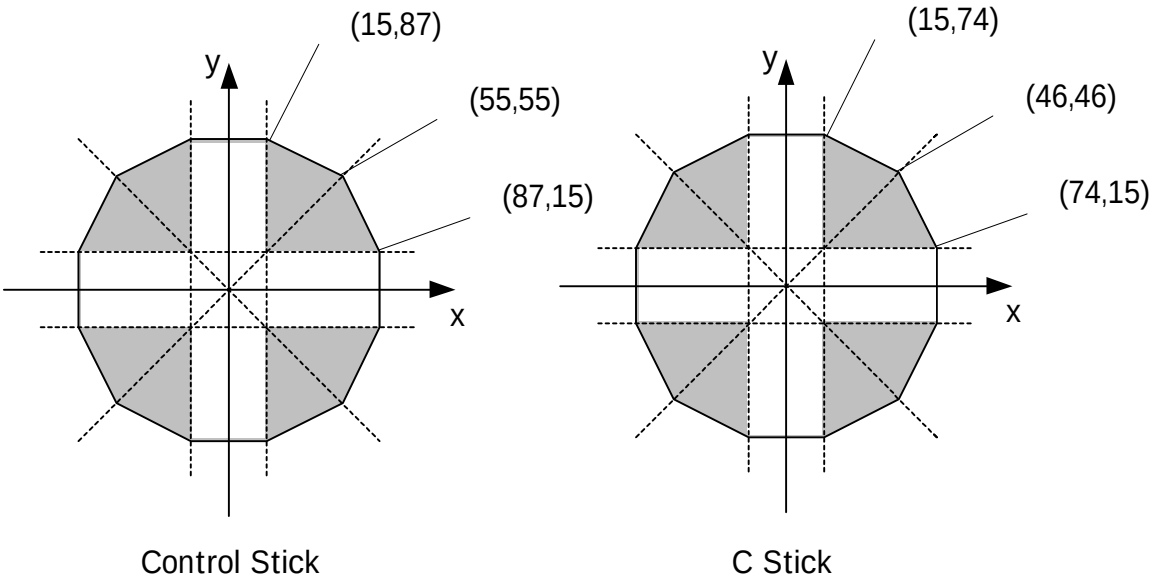

Figure 1. Usable range of input values

For the Control Stick, the unusable area is the part within the square defined by the four points (x, y) = (±15, ±15). Adjust the perpendicular component of the operation within the elongated plus-shaped region that extends along the X and Y axes to the points x, y = ±87. Finally, perform a clamp on values outside of the twelve-sided polygon that is defined by connecting the eight vertices of the plus-shaped region (x, y = ±87) and the vertices that are a distance of 77.8 (x = ±y) from the center.

Similarly, for the C Stick, the unusable area is the square defined by the four points (x, y) = (±15, ±15). For this stick, the perpendicular component of the operation is adjusted within the elongated plus-shaped region defined by the eight points that extend along the X and Y axes to the points x, y = ±74. Perform a clamp on values outside of the twelve-sided polygon that is defined by connecting these eight vertices and the vertices that are a distance of 65.1 (x = ±y) from the center.

The function PADClamp has been prepared to clamp values that are input outside the ranges described above and to adjust values that are input in the unusable area. (See Figure 2)

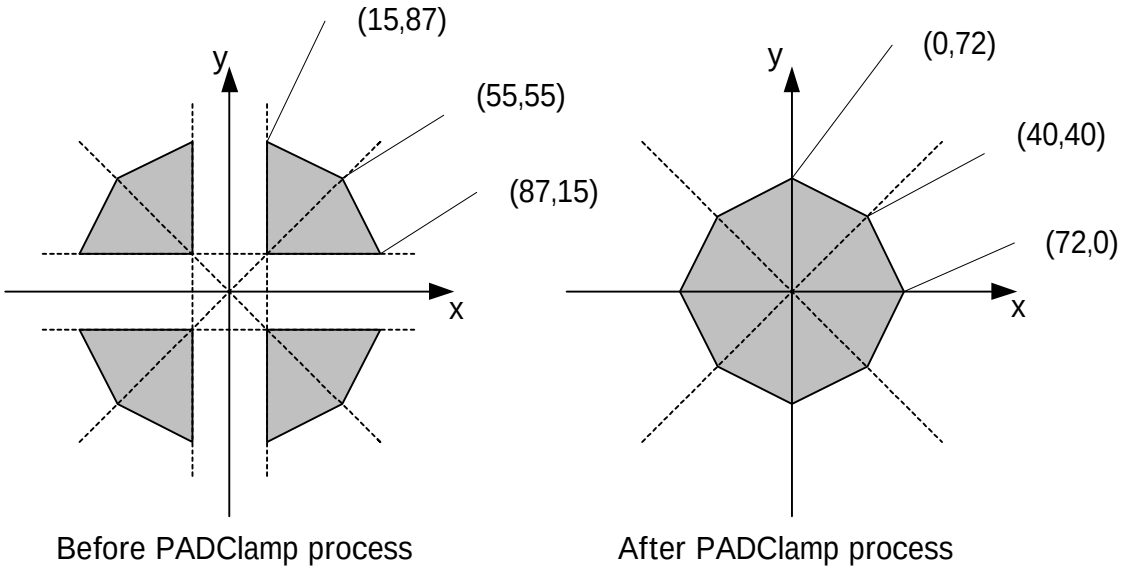

Figure 2. Usable range of input values after PADClamp

1.3.4.2 R Button and L Button

For the R Button and L Button, set the unusable area (from 0 to 30) and clamp the outside part (from 180 to 255).

1.4 Using the Controller to reset the application

If you are going to create an application that can be reset using an input from the Controller, either use an operation that involves pressing the B Button, X Button and START/PAUSE Button simultaneously for more than 0.5 second (Reset Command), or execute the reset from a menu selection inside the application. Do not use the Reset Command for other functions.

1.5 Rumble Feature

1.5.1 Enabling and disabling the Rumble Feature

If you build a game that uses the rumble feature, you must provide a way to enable and disable the feature. The reason for this is that certain players must not use this rumble function. This includes persons with osteoporosis and persons with injuries such as; bone fractures, dislocations, muscle tears or sprains in the finger, hand, wrist or arm.

1.5.2 Determining whether the Controller's Rumble Feature has been enabled or disabled

Even if the vibrations are necessary for a game, there may situations in which the user has disabled the Rumble Feature or is using a Controller that does not have the Rumble Feature (such as the WAVEBIRD). In such situations, you could provide an alternate representation of vibrations. You should be able to determine whether the Rumble Feature has been disabled. You can also use the returned values from the PADRead function to determine whether the Controller has the Rumble Feature.

1.6 Function Return Values

PAD_ERR_TRANSFER means that the corresponding Controller Socket data sampling failed. This is not the error value for when the Controller has been removed. When the Controller being used is removed, PAD_ERR_NO_CONTROLLER is returned. Therefore, do not create any application which will be recognized as the Controller being removed when PAD_ERR_TRANSFER is received.

2. Nintendo GameCube™ Wireless Controller (WAVEBIRD)

This section contains items of concern to those developing programs compatible with WAVEBIRD.

2.1 A warning about divulging information

When a player enters information into a WAVEBIRD, there is a possibility of that information being picked up by another waveless receiver. If you are going to prompt players to enter important information that relates to their privacy and property rights, you should display a warning about the danger of entering such information using a WAVEBIRD.